UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/215,788	12/21/1998	JERRIE L. COFFMAN	21936435X00	1611

7590 02/12/2003

HUNG H. BULL
ANTONELLI, TERRY, STOUT & KRAUS
1300 N. 17TH STREET
SUITE 1800
ARLINGTON, VA 22209

EXAMINER	
PRIETO, BEATRIZ	
ART UNIT	**PAPER NUMBER**
2142	

DATE MAILED: 02/12/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Paper No. 21

Application Number: 09/215,788
Filing Date: December 21, 1998
Appellant(s): COFFMAN ET AL.

MAILED
FEB 1 2 2003
Technology Center 2100

Hung H. Bui

For Appellant

Examiner's Answer

This is in response to appellant's brief on appeal filed 12/06/02.

(1) Real Party in Interest

A statement identifying the real party of interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences, which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments

The appellant's statement of the status of amendments contained in the brief is correct. Amendment after-final filed concurrently with appeal brief, consisting solely of the cancellation of claim 22, has been entered.

(5) Summary of the Invention

The summary of the invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Appellant's brief includes a statement that claims (1-21 and 23-28) of the following groups of claims stand or fall independently of each other and proved reasons as set forth in 37 CFR 1.192 (c) (7) and (c) (8).

Claims 24-28, for the purposes of examination with not stand individually for reasons stated in response to argument item (13). Claims 23-28 fall together (i.e., that they not are separately patentable).

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix A to the brief is correct.

(9) Prior art of record

The following is a listing of the prior art of record relied upon in the rejection of claims under appeal.

Heil et. al.	U.S. Patent No.	6,173,374	01-2001

Intelligent I/O (I_2O) Architecture Specification (Specs), version 1.5, March 1997.

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims: Claims 1-21 and 23-23 are presented for examination.

1. The following is quotation of 35 U.SC. §103(a), which forms the basis for all obviousness rejection, set forth in this Office action:

> (a) A patent may be not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art of record are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Heil et. al. (Heil) U.S. Patent No. 6,173,374 in view of Intelligent I/O Architecture Specification (Specs), version 1.5, March 1997.

Regarding claim 1, Heil teaches an access module (input/output platform (IOP)) (117 of Fig. 1) installed in a host system (150 of Fig. 2), (host system node(s) col 10/lines 25-58, col 6/lines 34-47, 60-64, a memory, I/O devices and a processor for controlling I/O devices (118 and 123), i.e. an "I/O platform access module"), said IOP access module comprising:

a host software driver (240 and 250 of Fig. 2) (Local Transport) arranged to provide an interface to an input/output platform (IOP) (col 10/lines 57-58) supporting an array of input/output devices (118) (col 10/line 66-col 11/line 35, I/O shipping, i.e. interface to the IOP (260) device driver layer);

a host software driver (270 and 280 of Fig. 2) (Remote Transport) arranged to provide an interface to said another system (151) (col 11/lines 5-11, 36-46), via a network data communication medium (121) (data network) (col 9/line 11-17) which provides input/output device access between the host system (host system node 150) and another system node (host system node 151) (col 8/lines 24-49); and

a host software driver (I/O Shipping 280 of Fig. 2) (a Connection Manager) arranged to establish connection services with remote servers on said computer network and create a communication between the host software driver (250 of Fig. 2) and host software driver (240 of Fig. 2) to provide access to the local or remote storage devices (col 11/lines 35-65, col 4/lines 58-61, and abstract);

however, Heil teachings for creating a communication path between the host software driver (250) and host software driver (240) to provide access to the local storage devices, is not called a "direct call path";

Specs teaches establishing connection services, and creating a direct call path between the Local transport and Remote transport to provide access to I/O devices; Specs teaches an host OS service module (OSM) (Local Transport) arranged to provide an interface to an IOP (see OSM and I_2O definitions on page 1-10), IOP supporting an array of I/O devices (see IOP definition, and sec. 2.1.1, on page 2-1); I_2O communication layer is the messaging layer, the messaging layer provides communication service from one module to another (sec 2.1.3, page 2-4); message services creates connections (sec. 4.1, page 4-1), establish a connection path between one IOP system and another remote IOP system (sec 4.5.4, page 4-68).

It would have been obvious to one ordinary skilled in the art at the time the invention was made to enable means for creating a direct call path between the driver modules to provide access to the local storage device, as taught by Specs, motivation would be enable a direct message passing between software driver layer for a particular class of I/O, such as those further specified by I_20 standards such as LAN ports, Ethernet or Token ring controllers, SCSI ports, etc providing an architecture that is operating-vendor-independent and adapts to existing system, creating scalable drives from high-end workstations to high-end server, as taught by Specs.

Regarding claim 2, IOP access module is one software module provided on a tangible medium (Heil: col 10/lines 28-50).

Regarding claim 3 and 8, containing limitations discussed on claims 1 and 14 and further, wherein said host system corresponds to a host server (Heil: abstract, said another system corresponds to any one of a remote servers, via data network (Heil: host computer system nodes on a server cluster, abstract, system interconnected via a data network, col 9/lines 11-17)

Regarding claim 4, one input/output processors (spec. page 1-1); one storage device as said input/output devices (Heil: (118), col 11/lines 7-35); a device driver module arranged to interface with said storage device (Heil: (260) col 7/line 7-12, 28-35); a communication layer which defines a mechanism for communications between the driver 250 (Local Transport) and the device driver module (260) (Heil: col 11/line 12-27, specs: messaging communication layer, sec. 2.1.3, page 2.4, Fig. 2.3)

Regarding claim 5, wherein said communication layer is responsible for managing all service requests (Spec: section 2.1.3, page 2.4, Fig. 2.3) and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages (Spec: API provide transport and messaging services, page 2.5).

Regarding claim 6, said communication layer comprises a message layer, which sets up a communication session (Spec: section 2.1.6, page 2-18), and a transport layer, which defines how information will be shared (Heil: coordinate retrieval of shared information, col 10/lines 66-col 11/line 11).

Regarding claim 7, containing limitation(s) substantially the same as claims 1 and 14, therefore same rational of rejection is applicant, claim 7, further includes;

a host driver module providing an interface to an input/output platform (IOP) supporting an array of storage devices; (Specs teaches an host OS service module (OSM) (Local Transport) arranged to provide an interface to an IOP (see OSM and I_2O definitions on page 1-10), IOP supporting an array of I/O devices (see IOP definition, and section 2.1.1, on page 2-1, section 2.1.4.1 on page 2-11);

a communication layer which defines a mechanism for communications between the system driver layer and the device driver layer (see Specs: I_2O communication layer (messaging service layer), which delivers I/O transactions messages from one software module to another, any where in the I_2O domain, independent of the modules hierarchy, section 2.1.3, page 2-4, Fig. 2-3), host system including a processor including operating system (Heil: 100 of Fig. 2, host device driver module operating system (OSM), col 10/lines 43-50 and a host system operating system col 10/lines 28-36).

Regarding claim 8, wherein said host computer node system corresponds to a host server, said remote systems corresponds to remote servers arranged in a cluster (system area network), and said computer network corresponds to a system area network for communications between said host system and said remote systems within said cluster (Heil: col 9/lines 11-17).

Regarding claim 9, wherein said communication layer is responsible for managing all service requests (Spec: section 2.1.3, page 2.4, Fig. 2.3) and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages (Spec: messaging is an interface between the modules, this interface is a set of APIs that provide transport and message services, page 2-5).

Regarding claim 10, wherein said communication layer comprises a message layer which sets up a communication session (Specs: section 2.1.6, page 2.18), and a transport layer, which defines how information will be shared (Heil: coordinate retrieval of shared information, col 10/lines 66-col 11/line 11).

Regarding claim 11, wherein said system driver module and said device driver module constitute a single device (Heil: col 10/lines 43-47) that is portable across a plurality of operating systems and host network platforms (Spec: section 1.2, page), and works interoperably with a plurality of storage devices and operating systems (Spec: coexist across different platforms, page 1-4; portability across platforms, section 2.5.2.1, on page 1-40).

Regarding claim 12, wherein said system driver module and said device driver module operate in accordance with an Intelligent Input/output (I_2O) specification for allowing storage devices to operate independently from the operating system (Heil: col 10/lines 28-42, abstract).

Regarding claim 13, wherein said driver module is a software module provided on a tangible medium (Heil: col 10/lines 28-50).

Regarding claim 14, substantially the same as claim 1, discussed above same rationale of rejection is applicable, further herein, the "IOP access module" discussed above (claims 1-7), is now a "system driver", and the "host system" in now a "host server on a computer network", and the "another system" is now "a remote server computer" wherein the prior art teaches a host driver configuration (117) of a host

server node (150) of a server cluster (area network) supporting I/O shipping to another remote node (151) (col 10/lines 28-58, abstract) comprising: an host driver (input/output platform (IOP)) arranged to control an array of local storage devices (118) (Heil: col 6/lines 34-47, 60-64, controlling access to I/O devices: col 10/lines 28-36, 51-58, host computer on a server cluster col 11/lines 1-27).

Regarding claim 15, wherein said IOP supports one I/O processor (Spec: Fig. 2.1 on page 2.1), and comprises: a device driver module (260) which interfaces the local storage devices for controlling said array of local storage devices (Heil: col 11/lines 12-35); and a communication layer which defines a mechanism for communications between the system driver module and the device driver module (see Specs: I_2O communication layer (messaging service layer), which delivers I/O transactions messages from one software module to another, section 2.1.3 on page 2-4, Fig. 2-3).

Regarding claim 16, wherein said communication layer is responsible for managing and dispatching all service requests (see Specs: section 2.1.3 on page 2-4, Fig. 2-3 communication layer is a network of object instance) and providing a set of APIs for delivering messages along with a set of support routines that process the messages (see Specs: messaging is an interface between the modules, this interface is a set of APIs that provide transport and message services, page 2-5), and is comprised of a message layer which sets up a communication session (see Specs: section 2.1.6 on page 2.18), and transport layer which defines how information will be shared (Heil: coordinate retrieval of shared information, col 10/lines 66-col 11/line 11).

Regarding claim 17-18, these claims are substantially the same as claims 11-12, discussed above, same rational of rejection is applicable.

3. Claims 19-21 and 23-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Heil U.S. Patent No. 6,173,374 in view of Intelligent I/O (I_2O) Architecture Specification (Specs) in further view of Bonola U.S. Patent No. 6,321,279.

Regarding claim 19, prior art teaches wherein said software driver (270, 280) (Remote Transport) prepares to accept requests from a computer node (151) of a remote server on a server cluster through said computer network (reception of request remotely generated for local data to be exported, col 11/lines 36-44, remote computer nodes of a server cluster, abstract), and

establishes a network management communication channel through the software driver (270, 280) (Remote Transport) (Heil: col 11/lines 54-65, export via software driver (240), col 10/lines 65-col 11/line 17), which waits for an external connection (Specs: table 4.5 on page 4-16, waiting for response) from said remote server on said computer network for exporting local device access onto said computer network using said direct call path between the Local Transport and the Remote Transport (Specs: section 1.1.2.2 on page 1-4, functions calls direct messaging passing between driver modules, for accessing I/O devices, page 6-4);

upon initialization said Local Transport scans the local bus so as to located and initialize all local input/output platforms IOPs and builds an list (opaque context) structure for each IOP (Specs: teach building a list of designations for each IOP, wherein after the IOP loads, it executes its initialization sequence, causing the IOP to scan its physical adapters, load and initialize appropriate devices, and build a logical configuration table (section 2.1.7.2 on page 2-25), located IOP are added to the system configuration table, host then gives each IOP a list of all IOPs (section 2.1.4.1 on page 2-11), said table lists all the IOPs registered devices and their availability, (item 8 on page 4-65); however the above mentioned prior art does not teach determining the number of IOP;

Bonola teaches means for determining upon initialization and a starting process, the number of CPU's present in the computer system along with the context of the computer system, col 9/lines 55-58, teaching means for querying so as to determine the number of input/output processing (IOP), col 8/line 5-13, 58-64);

It would have been obvious to one ordinary skilled in the art at the time the invention was made to modify existing teachings with means for determining the number of IOP as taught by Bonola motivation would be to prevent error in the installation phase determining the actual of detected processor, utilizing said number and the context information to initialize the drivers, supporting the allocation of shared memory without requiring extra memory, as in the prior art.

Regarding claim 20, wherein said Local Transport further has a send handler function and said Remote Transport further has a receive handler function which are respective program interfaces for receiving an inbound message from a remote server on said computer network for direct access to local input/output platform and for delivering an outbound message to said remote server on said computer network (Specs: inbound/outbound message exchanges, section 2.1.4.1, and page 2-11); when communicating with the IOP, the IOP receives a request message to that TID, the IOP uses the message's Function code (function call), queues the request to the event queue, if the driver sends requests, it must register their reply handlers via an API function call; when a Driver sends a request, it uses the appropriate InitiatorContext

value from handles are placed in a structure and which returns an InitiatorContext value identifying that structure; when a reply is received, the reply handler are retrieved from a structure identified by the InitiatorContext field and then queues the event, (section 5.1.5 on page 5-4, section 5.2.6 on page 5-14); In this manner an IOP descriptor structure for each input/output platform (IOP) is built, which includes a table (exported) of function call pointers and the context required by the driver to communicate with the IOP; In this manner send/receive (request/reply) handlers are exchange for corresponding inbound/outbound messages between the local and remote IOP modules.

Regarding claim 21, wherein said Remote Transport further builds an IOP connection structure including at least an IOP descriptor pointer which refers to the IOP descriptor structure of the Connection Manager for making a direct call to the Local Transport through the receive handler function and the send handler function.

Specs teaches a connection message structure is used by an IOP to connect one of its drivers and a device registered on another IOP, IOPs using send/receive request messages, messages comprising descriptor structure (e.g. InitiatorDevice and TargetDevice) used to refer to the driver and IOP that will send requests and the device on IOP that will receive them, section, page 4-20, said descriptor identifies the driver and the IOP, adapters and device types, section 5.2.1, page 5-7; IOP descriptor structure for each input/output platform (IOP) is built, which includes an exported table of function call (send/receive handler) pointers and the context required by the driver to communicate with the input/output platform (IOP); connection using said direct call path between the driver 250 and driver 240, discussed above in claims 1 and 14; wherein said connection request and its reply convey information enabling the two (IOPs) to establish a direct path for exchanging messages, section 2.1.4.1, on page 2-11.

Regarding claim 23, this claims combined limitations of claims 1-7, 14, and 19, same rationale of rejection applied to the limitation of theses claims are applicable to claim 23, further

initialization of said driver module, which provides access to a local storage system (Specs section 2.1.7.2 on page 2-25, section 2.1.4.1 on page 2-11, item 8 on page 4-65, Bonola: col 9/lines 55-58, col 8/line 5-13, 58-64) while bypassing protocol stacks 250/260of a host operating system (Heil: col 10/lines 28-65);

said driver module provides direct access to the local storage device system (118) (Heil: 240/250 of Fig. 2, col 4/lines 3-20, col 10/lines 57-58, col 10/line 66-col 11/line 35, (Spec: see OSM & I_2O definitions on page 1-10, OSM definition on page 2-3, IOP definition on page 1-10),

interfaces to other nodes of said system area network cluster (Heil: col 4/lines 21-29, col 11/lines 5-11, 36-46, col 9/line 11-17, col 8/lines 24-49, Spec: Fig. 2.4, transport services on page 2-5, section 5.2.7 on page 5-14, see transport layer Fig. 2-4, page 2-5),

providing connection services and coordinating functions responsible for creating a direct access (call path) between the Local Transport and the Remote Transport (see OSM and I_2O definitions on page 1-10, see IOP definition, and sec. 2.1.1, on page 2-1, section 2.1.3 on page 2-4, section 4.1 on page 4-1, section 4.5.4 on page 4-68);

searching (scanning) the local bus to locate and initialize all local input/output platforms (IOPs) (Heil: col 12/lines 9-col 13/line 3, Specs section 2.1.7.2 on page 2-25, list of all IOPs section 2.1.4.1 on page 2-11), and building a map (an IOP context structure) for each input/output platform (IOP) found (Heil: col 11/lines 53-col 12/line 59, descriptor structure col 13/line 5-15);

reception of request for a service connection from said remote server on said system area network (Heil: col 10/lines 28-65, col 11/lines 36-44);

determines the number of input/output platforms (IOPs) (Specs, built list of all IOPs section 2.1.4.1 on page 2-11, Bonola, number of IOPs col 9/lines 55-58, col 8/line 5-13, 58-64), and

building a descriptor structure for each input/output platform (IOP) which includes an exported table of function call pointers (Heil: col 13/lines 5-15) and the context required by the Local Transport to communicate with the input/output platform (IOP) (Spec section 5.1.5 on page 5-4, section 5.2.6 on page 5-14, pointers, table on Figure 3.36 on page 3-40, used to communicate, Spec on page 3-40); and

establishing a connections (system area network management communication channel) through the Remote Transport, which receives for an external connection from said remote server on said system area network for exporting local device access onto said system area network using said direct access (call path) between the Local Transport and the Remote Transport (Heil: col 11/lines 35-65, col 4/lines 58-61, and abstract, Spec: direct call path, see OSM and I_2O definitions on page 1-10, see IOP definition, and sec. 2.1.1, on page 2-1, section 2.1.3, page 2-4, creates connections section 4.1, page 4-1, connection path between one IOP system and another remote IOP system, section 4.5.4 on page 4-68).

Regarding claim 24, this claim comprises limitations addresses on claims 1-4, 7 and 19, same rationale is applicable, and further wherein the Local Transport is called herein "system driver module", further prior art a driver module (Heil: 117 of Fig. 1) which interfaces the local storage devices (118) which controls an array of local storage devices (Heil: processing I/O request, control the retrieval, and access, col 10/lines 57-col 11/line 27, supporting an array of input/output devices (118), col 10/line 66-col 11/line 35); a layers (240, 250, 260, 270) define communication between the system driver module and the

device driver module (Heil: communication between (240), (250) and device driver module (260), col 11/line 1-46, Specs: messaging communication layer, sec. 2.1.3, page 2.4, Fig. 2.3).

Regarding claim 25, this claims contains the combined limitations of claims 5-6, and 9-10, discussed above, same rationale of rejection is applicable.

Regarding claim 26, this claim contains the combined limitations of claims 11-12, discussed above, same rationale of rejection is applicable.

Regarding claims 27-28, these claims are substantially the same as claims 19-20 discussed above, same rationale of rejection is applicable.

(11) Response to Argument

1. Applicant argues in regards claims 1-18 (a), that neither of the references teach any module called "Local Transport", "Remote Transport" and "Connection Manager" as "part of a host driver module - an upper module of a driver system", as expressly identified in each claims 1, 7 and 14.

In response to argument (a), that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *a module as part of the host driver module-upper module of a drive system*") are not recited in the rejected claim(s) 1, 7 and 14. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). In this case, claims 1, 7 and 14, **do not recite, nor do not expressly identify** "*a host driver module-an upper module of a driver system*".

claim 7 recites a "a host driver module arranged to provide an interface to the operating system":

Prior art teaches a host bus adapters (HBA) which adapts, connects a host computer system to an I/O device for I/O shipping of block level requests, adapting the requests for exchange to the I/O device (Heil: col 3/lines 64-col 4/line 20), processing I/O request from the host system (Heil: col 4/lines 41-42) and functioning as a host bus driver for controlling access to I/O devices (Heil: col 6/lines 65-67).

Additionally, prior art teaches a host driver module arranged to interface to the operating system (Spec: section 1.1.2 on page 1-2, see OSM and I_2O definitions on page 1-10).

Therefore prior art teaches a host driver module arranged to provide an interface I/O request from a host computer node system executing on an operating system.

2. Applicant argues in regards to claims 1, 7 and 14, (b) that office action has failed to establish a prima facie case of obviousness, because there is not factual evidence to support such a conclusion of obviousness. Specifically, because the prior art's host bus adapter does not constitute a host driver module, nor does Heil disclose any host driver installed in a host system to provide access to I/O devices, nor does the prior art disclose where the HBA are part of any host operating system.

In response to argument (b), that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *"a host driver installed in a host system to provide access to I/O devices within the host driver is part of an host operating system" nor "a host system (as part of a host operating system) to access an input/output platform (IOP)"*) are not recited in the rejected claim(s) 1, 7 and 14. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). In this case, claims 1, 7 and 14, **do not recite, nor do not expressly identify** *"a host driver module-an upper module of a driver system being part of an operating system"*, **nor** "a host driver module installed in a host system *(as part of a host operating system)* to access input platform (IOP)", as argued. It is respectfully noted the applicant's specification do not contain an explicit recitation of a "*host system*", or "data network" nor mentioned of the terms "*host system*" or "data network" (see MPEP §2126), therefore the term is given the broadest reasonable interpretation inlight of the specifications as mandated (see MPEP §2181).

claim 1 recites is a "an access module (an input/output platform (IOP) access module) installed in a host system for providing input/output device access between the host system and another system, via a data network";

Prior art teaches host bus adapter driver which reside on a host computer nodes (host systems) of a server cluster (abstract), each host adapter driver adapts, connects (interface) a host computer system to an I/O device (118, 123) for I/O shipping of block level requests (Figs. 1-2), processing the I/O requests for access to the I/O devices (col 3/lines 64-col 4/line 20, col 6/lines 34-47, 60-64), between the host computer system (150) and another host computer system node (151) via a network infrastructure for data transports (data network) between nodes in a server cluster (120 of Fig. 2) (col 8/lines 12-59) for controlling I/O devices (118 and 123) (col 10/line 66-col 11/line 27);

Additionally, Specs teaches a driver module installed in a host system for providing I/O device access (Spec: split driver model can execute on operating system environment: section 2.1.2 on page 2-2), Fig. 2.2 on page 2.3 illustrates an OSM driver module, and the messaging layer module executing on a host, to provide interface to the operating system: section 2.1.2 on page 2-3. The OSM driver module and the host MessengerInstance (Messaging Layer) are host-specific, i.e. execute in a specific host OS

environment: section 2.1.3.3 on page 2.9, messaging layer running on a particular platform: see MessengerInstance definition on table 1-7 on page 1-10).

3. Applicant argues in regards to claims 1, 7, and 14, (c) that prior art does not teach a "Local Transport arranged to provide an interface to an I/O platform supporting an array of I/O devices", because neither (230) nor (250) software layer driver modules can be reasonable interpreted to meet the claim limitation.

In response to argument (c), Heil teaches where host software driver (240 and 250 of Fig. 2) (Local Transport) is arranged to provide an interface to an input/output platform (IOP) (260) (col 10/lines 43-47, specific device driver) accessing (supporting) an array of I/O devices (col 7/lines 57-61, I/O devices 118), wherein the software layer 240 using 250, processes and redirect I/O request, i.e. interfacing, (col 10/line 66-col 11/line 35), I/O shipping i.e. interface to the IOP (260) device driver layer for accessing local I/O storage devices (118); therefore prior art teaches host software driver 240/250 (Local Transport) arranged to provide an interface to an I/O platform supporting an array of I/O devices.

Additionally, Specs teach a host driver module ("Local Transport") arranged to interface to an input/output platform IOP (Spec: see OSM & I_2O definitions on page 1-10, OSM definition on page 2-3), supporting an array of input/output devices (Spec: see IOP definition on page 1-10);

4. Applicant argues in regards to claims 1, 7, and 14 (d), that prior art does not teach "a Connection Manager" as claimed, because (i) prior art's software layer (280) is not part of the host driver module-upper module which is host OS-specific and installed as part of an operating system of a host server, nor further teaches according to applicant (ii) a connection manager is arranged to create a direct call path between the Local and Remote Transports, and according to applicant "the creation of a direct call path between the Local and Remote Transport is essential in terms of providing direct access to I/O storage devices with out incurring the overhead of the OS and does not require the search directory functions of the prior art".

In response to argument (d), that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "*a host driver as part of the host driver module-upper module which is host OS-specific and installed as part of an operating system of a host server, nor a direct call path between the Local and Remote Transport which provides direct access to I/O storage devices with out incurring the overhead of the OS and does not require search directory functions*") are not recited in the rejected claim(s) 1, 7 and 14. Although the claims are interpreted in

light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Claim limitation which applicant makes reference recites, "a Connection Manager arranged to establish connection services and to create a direct call path between the Local Transport and the Remote Transport so as to provide access to input/out devices";

Heil teaches a host software driver (280 of Fig. 2) (a Connection Manager) arranged to establish connection services with remote servers on the computer network (establish/maintain communication with another: col 4/lines 15-20, 58-61, local/remote: col 4/lines 41-44) and create a communication between the host software driver (250 of Fig. 2) (Local Transport) and host software driver (270 of Fig. 2) (Remote Transport) to provide access to the local or remote storage devices; whereby software driver (270) manages the transmission of locally generated request for remote data (e.g. 123) out of the network and the remotely generated request for local data (e.g. 118) (col 11/lines 36-44) which are communicated to (240) which determines if the request of to be satisfied locally (using 250, col 11/lines 11-27) or remotely (using 280 I/O shipping, col 11/lines 54-col 12/line 7), managing and coordinating the retrieval of the data (col 11/lines 1-10), although Heil teaches the software driver (240 and 250) and driver (270 and 280) communicate to provide access to the local data residing in the local I/O devices or remote data residing on another system, it does not explicitly teach that the communication between these software drivers is a "direct call path".

Aditionally, Specs teaches a message-based interface which enables direct messages passing between any two device driver modules (section 1.1.2.2 on page 1-4); when an IOP wants to connect to another IOP, the connection request and its reply convey information enabling the two IOPs to establish a direct path for exchanging messages, (section 2.1.4.1 on page 2-11); message-based interfaces provide direct communication between the OSM and a DDM (section 2.1.3.3 on page 2-9);

Further Specs additionally teach, a host driver module ("Local Transport") arranged to interface to an input/output platform IOP (Spec: see OSM & I_2O definitions on page 1-10, OSM definition on page 2-3), supporting an array of input/output devices (Spec: see IOP definition on page 1-10);

a Transport layer ("Remote Transport) arranged to interface to another system, enabling transport services for interfacing (accessing & transporting data) to another system (see Fig. 2.4, transport services on page 2-5, transport services provide direct memory access (DMA) capability between system memory and local system memory, section 5.2.7 on page 5-14, see transport layer Fig. 2-4, page 2-5).

a messaging layer provides the service to establish, use and tear down communication channels (connections) between modules (section 2.1.6 on page 2-18), supporting connection with an IOP on another system (communication between modules on different IOPs, on page 1-11); a Connection

Management enabled by the Messenger service provides establishing (setup) a connection a local and remote IOP (section 2.2.3.3.9 on page 2-31);

Connection Management (Connection Manager) enabled by the message services (message-based interface) providing connection services and creating a direct call path, as discussed above, between the OSM ("Local Transport") and the Transport Layer ("Remote Transport), (see Figure 2.7 on page 2-10).

5. Applicant argues in regards to claims 1, 7 and 14, (e) that the prior art of records provides an unclear motivation to one ordinary skilled in the art enables the claim limitation.

In response to applicant's argument (e) that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, the teachings of the secondary reference further exemplify standard technology (enabling methodology) explicitly referenced as used on the primary reference, therefore explicit suggestion in the primary reference motivates one ordinary skill in the art to modify the teachings using the teachings of the secondary reference to produce the claimed invention (discussed above), and provides evidence indicating the modification would be successful.

6. Applicant argues in regards to claims 2-6, 8-13 and 15-18, (f) specifically claims 4 and 15, that the prior art does not teach claim an I/O processor, an I/O storage device, a device driver to interface said storage device and a communication layer which defines a mechanism for communicating between the Local Transport and device driver module, because according to applicant prior art does not teach a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices.

In response to applicant's argument (f) that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. "*a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices*") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

7. Applicant argues in regards to claims 5-6, 9-10, and 16, (g) that the prior art of record does not teach the claim limitation on these claims, because the prior art does not disclose a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices, wherein the reference Heil does not disclose any host driver module nor any IOP as defined on claims 4 and 15, and the Specs reference does not teach a communication layer as defined on claim 5.

In response to applicant's argument (g) that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. *"a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices"*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Prior art teaches claim 4, which reads, an one input/output processors (Specs: page 1-1); one storage device as said input/output devices (Heil: col 11/lines 7-35, I/O devices 118); a device driver module arranged to interface with said storage device (Heil: col 7/line 7-12, 28-35, driver 260); a communication layer which defines a mechanism for communications between the driver 240/250 (Local Transport) and the device driver module (260) (Heil: 240 using (250) manages and coordinated the retrieval of data using (260), col 10/lines 66-col 11/line 27, that retrieve data from (118), and specs: messaging communication layer, sec. 2.1.3, page 2.4, Fig. 2.3); and

Prior art teaches claim 5, which reads, said communication layer is responsible for managing all service requests (Spec: section 2.1.3 on page 2.4, Fig. 2.3, communication layer is the messaging layer for providing communication services) and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages (Spec: API provide transport and messaging services, page 2-5, message-based interface includes an API which consisting of function calls, section 2.1.3.2.1 on page 2-7).

8. Applicant argues in regards to claim 11 (h), that the prior art of record does not teach claim limitations, because prior art does not teach "a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices", and there is not possibility of any incorporation of both the "host driver module" and "device driver module" as defined on claim 11.

In response to applicant's argument (h) that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. *"a host driver-upper host OS specific module that interfaces a host operating system that is separate and distinct from a device driver module a lower device-specific module that interfaces I/O devices"*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Prior art teaches claim 11, which reads wherein said system driver module and said device driver module constitute a single device (Heil: each HBA contains 240, Local ISM 250 and HDM 280 and Remote ISM 270 and HDM 280 drivers, col 10/lines 66-col 11/line 27, Fig. 2, One module device driver having OSM, ISM and HDM stackable drivers, col 10/lines 43-57) that is portable across a plurality of operating systems and host network platforms (Spec: section 1.2, page), and works interoperably with a plurality of storage devices and operating systems (Spec: coexist across different platforms, page 1-4; portability across platforms, section 2.5.2.1, on page 1-40).

9. Applicant argues in regards to claims 19-21 and 23-28, (i) prior art does not teach claim 19, specifically, "a host driver module" including a "Local Transport", "a Remote Transport" and "a Connection Manager" "that are separate and distinct forming a device driver module a lower [device-specific] module installed in IOP that interfaces I/O devices"

In response to applicant's argument (i) that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. *"host driver module" including a "Local Transport", "a Remote Transport" and "a Connection Manager" "that are separate and distinct forming device driver module a lower [device-specific] module installed in IOP that interfaces I/O devices"*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

10. Applicant argues further argues in regards to claims 19-21 and 23-28, (j) that prior art does not teach claim 19, specifically, claim limitation, "upon initialization, said Local Transport scans the local bus so as to locate and initialize all local I/O platforms and builds an opaque "context" structure for each IOP".

Specs: teach building a list of designations for each IOP, wherein after the IOP loads, it executes its initialization sequence, causing the IOP to scan its physical adapters, load and initialize appropriate

devices, and builds a logical configuration table, (section 2.1.7.2 on page 2-25), located IOP is added to the system configuration table, host then gives each IOP a list of all IOPs, (section 2.1.4.1 on page 2-11), said table lists all the IOPs registered devices and their availability, (item 8 on page 4-65), for each device registered, the driver provides a list of message handlers, one for each message function, received messages handlers are used to builds a message dispatch table, (section 2.2.4.3 on page 2-32), each registered I_2O device provides message handlers that process the messages to the device, the identify of the event handler is accompanied by a context variable so that the message variable can determine the associated I/O transaction of each registered IOP, (section 2.2.4.1 on page 2-32).

Therefore prior art teaches upon initialization, scanning the local bus to locate and initialize all local I/O platforms and builds an opaque "context" structure for each IOP.

11. Applicant argues in regards to claims 19-23, and 23-28 (k), that prior art does not teach claim limitation, specifically, wherein said Connection Manager queries said Local Transport so as to determine the number of input/output platforms, builds an IOP descriptor structure for each input/output platform (IOP) which includes an exported table of function call pointers and context required by the Local Transport to communicate with the input/output platform (IOP);

Specs teaches a driver provides a pointer to a message dispatch table (IOP descriptor structure) for each input/output platform (IOP) when created and assigned a unique identifier (TID), this table contains functions codes and message handlers for processing request messages, used by the IOP and the device drive; a device driver must register these handles in the table via API function calls; a device driver when sending a message request, it uses the appropriate InitiatorContext value (context) returned when the message handle was registered in the table, (section 5.1.5 on page 5-4, section 5.2.6 on page 5-14, pointer to the each dispatch table for each registered device). Each IOP has a configuration table (Figure 3.36 on page 3-40), each IOP publishes this information in the logical configuration table, and the OSM (Local Transport) uses this information to determine which devices to query (page 3-40).

Additionally, Bonola teaches means for determining upon initialization and a starting process, the number of CPU's present in the computer system along with the context of the computer system, col 9/lines 55-58, teaching means for querying so as to determine the number of input/output processing (IOP), col 8/line 5-13, 58-64).

Therefore, prior art teaches building an IOP descriptor structure for each input/output platform (IOP) which includes an table (exported) of function call pointers and the context required by the software driver to communicate with the input/output platform (IOP). It is noted that the broadness of the claim language "the number of IOPs" does not preclude prior art teachings "generated list of all

existing/registered IOPs upon initialization", therefore the teachings of Specs meet the claim limitation as interpreted.

12. Applicant argues in regards to claim 25 (l), prior art does not teach claim, specifically, prior art does not teach building a descriptor structure for each IOP which includes an exported table of function call pointers and context required for communicating with the IOP;

In response to argument (l), prior art teaches building an IOP descriptor structure for each input/output platform (IOP) which includes an dispatch (exported) table of function call pointers and the context required by the software driver to communicate with the input/output platform (IOP): Specs: section 5.1.5, page 5-4, section 5.2.6, page 5-14, and Bonola: col 9/lines 55-58, and col 8/line 5-13, 58-64).

13. Applicant argues in regards to claims (24-28) (m), prior art does not teach claim limitation because the prior art nowhere teaches the features of the claim.

In response Applicant's arguments (m), arguments in regards to claims 24-28 fail to comply with 37 CFR §1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

It should be noted that 37 CFR 1.192(c)(7) requires the appellant to perform two affirmative acts in his or her brief in order to have the separate patentability of a plurality of claims subject to the same rejection considered. The appellant must (A) state that the claims do not stand or fall together and (B) present arguments why the claims subject to the same rejection are separately patentable. Where the appellant does neither, the claims will be treated as standing or falling together. Where, however, the appellant (A) omits the statement required by 37 CFR 1.192(c)(7) yet presents arguments in the argument section of the brief, or (B) includes the statement required by 37 CFR 1.192(c)(7) to the effect that one or more claims do not stand or fall together (i.e., that they are separately patentable) yet does not offer argument in support thereof in the "Argument" section of the brief, the appellant should be notified of the noncompliance as per 37 CFR 1.192(d). Ex parte Schier, 21 USPQ2d 1016 (Bd. Pat. App. & Int. 1991); Ex parte Ohsumi, 21 USPQ2d 1020 (Bd. Pat. App. & Int. 1991). Applicant indicates that because claim 23 contains limitations similar to claim 19, same argued reasoning is applicable to claim 23, and further that claims 23 is patentable because its dependent claims 24-28 are separately patentable. Arguments regards claims 24-28 fail to comply with 37 CFR §1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the

claims patentably distinguishes them from the references. Hence, claims 24-28 fall together with independent claim 23.

(12) Conclusion

The ultimate determination of patentability must be based on consideration of the entire record, by a preponderance of evidence, with due consideration to the persuasiveness of any arguments and any secondary evidence. In re Oetiker, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). The submission of objective evidence of patentability does not mandate a conclusion of patentability in and of itself. In re Chupp, 816 F.2d 643, 2 USPQ2d 1437 (Fed. Cir. 1987). Facts established by rebuttal evidence must be evaluated along with the facts on which the conclusion of a prima facie case was reached, not against the conclusion itself. In re Eli Lilly, 902 F.2d 943, 14 USPQ2d 1741 (Fed. Cir. 1990). In other words, each piece of rebuttal evidence should not be evaluated for its ability to knockdown the prima facie case. All of the competent rebuttal evidence taken as a whole should be weighed against the evidence supporting the prima facie case. In re Piasecki, 745 F.2d 1468, 1472, 223 USPQ 785, 788 (Fed. Cir. 1984). Although the record may establish evidence of secondary considerations which are indicia of nonobviousness, the record may also establish such a strong case of obviousness that the objective evidence of nonobviousness is not sufficient to outweigh the evidence of obviousness. Newell Cos. v. Kenney Mfg. Co., 864 F.2d 757, 769, 9 USPQ2d 1417, 1427 (Fed. Cir. 1988), cert. denied, 493 U.S. 814 (1989); Richardson-Vicks, Inc., v. The Upjohn Co., 122 F.3d 1476, 1484, 44 USPQ2d 1181, 1187 (Fed. Cir. 1997) (showing of unexpected results and commercial success of claimed ibuprofen and psuedoephedrine combination in single tablet form, while supported by substantial evidence, held not to overcome strong prima facie case of obviousness).

For the reasons above it is believed that the rejection should be maintained.

Respectfully submitted,

B. Prieto
Patent Examiner
GAU 21$\not{4}$2

(Primary Examiner)

LE HIEN LUU
PRIMARY EXAMINER

(Conferee Examiner)

Antonelli, Terry, Stout & Krauss, LLP
1300 North Seventeenth St., Suite 1800
Arlington, VA 22209

DAVID WILEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100